Low rank prior in single patches for non-pointwise impulse noise removal

Wang, Ruixuan; Pakleppa, Markus; Trucco, Emanuele

Published in:
IEEE Transactions on Image Processing

DOI:
10.1109/TIP.2015.2400225

Publication date:
2015

Document Version
Peer reviewed version

Low rank prior in single patches for non-pointwise impulse noise removal

Ruixuan Wang, Markus Pakleppa, Emanuele Trucco

Abstract

This paper introduces a low rank prior in small oriented noise-free image patches: Considering an oriented patch as a matrix, a low-rank matrix approximation is enough to preserve the texture details in the optimally oriented patch. Based on this prior, we propose a single-patch method within a generalized joint low-rank and sparse matrix recovery framework to simultaneously detect and remove non-pointwise random-valued impulse noise (e.g., very small blobs). A weighting matrix is incorporated in the framework to encode an initial estimate of the spatial noise distribution. An accelerated proximal gradient method is adapted to estimate the optimal noise-free image patches. Experiments show the effectiveness of our framework in detecting and removing non-pointwise random-valued impulse noise.

Index Terms

Low rank prior, random-valued impulse noise detection and removal, joint low-rank and sparse matrix recovery, accelerated proximal gradient.

I. Introduction

This paper aims to remove random-valued impulse noise (RVIN) with varying sizes and irregular shapes (so called 'non-pointwise' RVIN, e.g., small particles suspended in water; see Section VI-F). Based on the observation that almost any optimally oriented (defined later) small,

R. Wang and E. Trucco are with the School of Computing, University of Dundee, UK. M. Pakleppa is with the School of Engineering, Physics and Mathematics, University of Dundee, UK.
E-mail: {ruixuanwang, manueltrucco}@computing.dundee.ac.uk; m.pakleppa@dundee.ac.uk

noise-free image patch (a matrix) can be approximated by a low-rank patch with texture details well preserved, we propose a single-patch method to simultaneously detect and remove RVIN within a generalized joint low-rank and sparse matrix recovery framework. The original matrix recovery framework has been recently used for image and video denoising [16], but it requires multiple similar image patches, with each patch vectorized as a column in the matrix. Such a multi-patch method often limits the size of image patches to be relatively small (e.g., 8×8 pixels) in order to more likely find multiple similar patches within a single image and to collaboratively and effectively detect and remove traditional (single-pixel) RVIN. Its performance degrades with non-pointwise RVIN, as shown in our experiments, even with noisy regions as small as 3×3 pixels (Section VI-D). Instead, our single-patch method completely avoids searching for similar patches, and importantly, uses larger-size patches (e.g., 40×40) to effectively detect and remove non-ponintwise RVIN.

One motivation of this study is to detect and remove particles suspended in water in hydro-colonoscopy video images (see Section VI-F), in which the size of particles varies and could be as large as 15×15 pixels. In this case, traditional methods to detect and remove single-pixel RVINs are much less effective compared to the proposed method. While image inpainting techniques [15] have been well explored to fill degraded or damaged (often large) image regions, these techniques assume that the regions to be inpainted (or filled) are manually provided in advance.

A. Related work

We briefly discuss related work on impulse noise detection and removal and low-rank matrix recovery; see [4] and [17] for recent, comprehensive reviews on denoising.

There are mainly two types of impulse noise: salt-and-pepper noise (black or white), and RVIN (any gray value). The median filter and its extensions are often effective to remove salt-and-pepper noise but can corrupt some textures (see results in Section VI-D). To reduce undesired corruption, various two-stage methods have been developed mainly to remove RVIN, first detecting the locations of noisy pixels, then recovering intensities only at noisy locations using certain filtering or variational methods, e.g., adaptive center weighted median filter (ACWMF) [6], neighboring similarity-based noise detector followed by a triangle-based linear interpolation for pixel restoration (NS-LI) [7] (see also [3][8]), rank-ordered absolute difference (ROAD) noise detector followed by a trilateral filtering [13], and a logarithmic version of the ROAD followed

by edge-preserving regularization (EPR) for pixel restoration (ROLD-EPR) [11]. NS-LI, ROAD-trilateral, and ROLD-EPR methods can preserve edges better than median-type methods like ACWMF as they consider local structures during pixel restoration. However, the accuracy of these two-stage methods depends crucially on the performance of the location stage. If noisy pixels cannot be detected correctly, e.g., when noise is structured rather than single-pixel, the overall noisy removal will be limited.

Given the excellent performance of non-local methods [4][9], learned sparse models [12][24], and the combination of both [10][23] for random Gaussian noise, they were explored for impulse noise as well [27][29]. Non-local methods use redundant visual information within an image (i.e., self-similarity) to group similar image patches together, followed by collaborative filtering [4][9]. Sparse methods also use redundant information by assuming each patch can be well approximated by a linear combination of a small subset of patches ('words') within a large dictionary. Both types of methods can preserve texture details very well. However, patch size is limited (e.g., 8×8 pixels) as it may become difficult to find multiple similar larger-size patches within an image for non-local methods, and to represent a larger-size patch by a linear combination of other patches for sparse methods. Crucially, impulse noise often needs to be detected first to reduce the effect of noisy pixels on patch matching and dictionary learning. Similar to the above two-stage methods, the overall accuracy of impulse noise removal is largely limited by the performance of the initial impulse noise location.

A joint low-rank and sparse matrix recovery framework was recently applied to detect and remove impulse noise simultaneously [16], using the self-similarity prior as in the non-local methods. In this approach, multiple similar patches are vectorized as columns in a matrix which is decomposed into a low-rank matrix, representing denoised similar patches, and a sparse matrix, representing the sparse set of impulse noise in patches. Gaussian noise is also modeled in the framework as a regularization term. The framework has proven effective in various applications, e.g., video denoising and inpainting [16], background removal and removing shadows and specularity from face images [5]. However, the method is limited by the usage of multiple similar patches and the small size of patches.

Unlike the above, this paper uses a different type of prior information, the 'low-rank prior', not for the whole image but for each appropriately oriented image patch. It has been observed that low-rank textures exist in image regions having deterministic regular or periodic patterns

[20][28]. In contrast, we found that almost any small (e.g., 10×10 to 40×40) image patch from images of natural or man-made objects or scenes with resolution of about 300×300 pixels (within our experiments, see Section 6), if rotated by a characteristic orientation defined later, has a low-rank approximation with texture details (including edges) well preserved (Section VI-B). In addition, compared to the closely relevant work [20], which used hard-thresholding to estimate the corrupted regions and applied a different optimization method to solve a more complicated optimization problem for different applications (e.g., image completion), our approach uses a soft-weighting matrix to weight each pixel and tries to denoise with a different but simpler (fewer constraints and variables) optimization problem. This paper is the extension of our recent conference publication [26].

II. Problem formulation

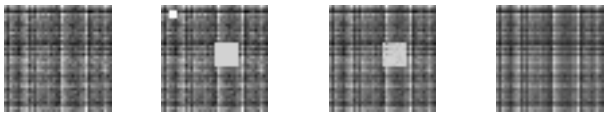
When a rectangular gray image patch $\mathbf{P}$ contains random Gaussian noise and RVIN, $\mathbf{P}$ may be decomposed as $\mathbf{P} = \mathbf{L}^* + \mathbf{S}^* + \mathbf{N}^*$, where $\mathbf{L}^*$ represents the unknown noise-free patch, $\mathbf{S}^*$ represents the unknown impulse noise, and $\mathbf{N}^*$ is a matrix of Gaussian noise [16].

$\mathbf{L}^*$ can be considered a low-rank matrix due to the low-rank prior for single patches (Section VI-B). Also, since the number of pixels corrupted by impulse noise is generally much smaller than the total number of pixels, $\mathbf{S}^*$ is a sparse matrix. As a result, the problem of image denoising can be formulated as an optimization problem [16], i.e., to minimize the following function $E_1(\mathbf{L}, \mathbf{S})$ over the matrix variables $\mathbf{L}$ and $\mathbf{S}$,

$$E_1(\mathbf{L}, \mathbf{S}) = \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 + \frac{1}{2\mu} \|\mathbf{P} - \mathbf{L} - \mathbf{S}\|_F^2 , \quad (1)$$

where $\|\cdot\|_*$ is the nuclear norm (i.e., sum of singular values) and considered as a convex relaxation of the function estimating the rank of a matrix; $\|\cdot\|_1$ is the sum of the absolute values of all matrix entries and considered as a convex relaxation of the sparsity measurement (i.e., number of non-zero entries) of a matrix; $\|\cdot\|_F$ is the Frobenius norm; and λ, μ are two regularization parameters.

While the above optimization framework has been used for image and video denoising [16], each image patch has been considered as a column in a matrix. Such a multi-patch method is subject to the limitations discussed in Section I. Now more specifically, using larger-size patches in the multi-patch method will generally make multiple patches less similar to each other and



(a) (b) (c) (d)

Fig. 1: Illustrative example of low-rank matrix recovery. (a) A synthetic clean image patch of size 40×40 pixels with rank 20. (b) A synthetic noisy patch by adding a smaller (3×3 pixels) non-pointwise RVIN at the top-left corner and a larger (9×9 pixels) one around the center. (c) The larger noise has been largely remained by minimizing Equation (1). (d) Both are removed by minimizing Equation (2).

hence lead to over smoothing of the current patch (Figures 9 and 10). Instead, our method requires no search as it considers a single patch as the matrix $\mathbf{P}$ and the patch size can be larger (e.g., 41×41) without decreasing the efficiency of matrix recovery. More importantly, using larger-size patches allows us to remove non-pointwise RVIN.

The limited ability of the optimization framework (Equation 1) to remove non-pointwise impulsive noise is another key limit we address. The minimization of $E_1(\mathbf{L}, \mathbf{S})$ will generally lead to relatively small value of the term $\|\hat{\mathbf{L}}\|_* + \lambda\|\hat{\mathbf{S}}\|_1$ (and $\frac{1}{2\mu}\|\mathbf{P} - \hat{\mathbf{L}} - \hat{\mathbf{S}}\|_F^2$ as well) at the estimated optimal solution, $\hat{\mathbf{L}}$ and $\hat{\mathbf{S}}$. However, if very different (e.g., with much higher intensity value than the signal) non-pointwise impulse noise exists, the true solutions $\mathbf{S}^*$ and $\mathbf{L}^*$ often bring about a much larger $\lambda\|\mathbf{S}^*\|_1$ (due to the set of higher impulse noise values) and modestly smaller $\|\mathbf{L}^*\|_*$ than those at the estimated solution $\hat{\mathbf{S}}$ and $\hat{\mathbf{L}}$. So the minimum may not correspond to the true solution, i.e., $E_1(\mathbf{L}^*, \mathbf{S}^*) > E_1(\hat{\mathbf{L}}, \hat{\mathbf{S}})$, and the non-pointwise impulse noise will remain, to some extent, in the estimated optimal signal $\hat{\mathbf{L}}$ (Figure 1c).

Hence we propose a generalized version of the optimization framework to denoise an image patch effectively in the presence of non-pointwise (multi-pixel) RVIN:

$$E_2(\mathbf{L}, \mathbf{S}) = \|\mathbf{L}\|_* + \lambda \|\mathbf{W} \circ \mathbf{S}\|_1 + \frac{1}{2\mu} \|\mathbf{W} \circ (\mathbf{P} - \mathbf{L} - \mathbf{S})\|_F^2 \tag{2}$$

where $\mathbf{W}$ is a soft-weighting matrix with each entry value in $[0, 1]$, and $\circ$ denotes the Hadamard (i.e., entry-wise) product of two matrices. When every entry in $\mathbf{W}$ is set to 1, $E_2(\mathbf{L}, \mathbf{S})$ becomes $E_1(\mathbf{L}, \mathbf{S})$.

$\mathbf{W}$ can encode the initially estimated spatial distribution of impulse noise in the image patch.

Initial estimates, obtained by any impulse noise detector, correspond to entries in $\mathbf{W}$ with values close to 0. Compared to the original optimization framework, Equation (1), the true solution, $\mathbf{S}^*$, will more likely correspond to a much smaller $\lambda\|\mathbf{W} \circ \mathbf{S}^*\|_1$, because at least part of the higher impulse noise values are counterbalanced by the corresponding smaller entry values in $\mathbf{W}$. From the regularization point of view, smaller entry values at initially estimated impulse noise locations in $\mathbf{W}$ will decrease the regularization effect of the second and the third terms in $E_2(\mathbf{L}, \mathbf{S})$, so that the entries at the corresponding locations in the matrix variable $\mathbf{L}$ can be searched in a larger feasible region in order to get a smaller $\|\mathbf{L}\|_*$. As a result, the minimum of $E_2(\mathbf{L}, \mathbf{S})$ will more likely correspond to the ground-truth solution $\mathbf{L}^*$. Therefore, $\mathbf{W}$ is expected to effectively help recover the corrupted signals (Figure 1d).

III. Optimization

The target function E_2 in Equation (2) can be minimized by accelerated proximal gradient (APG) [21][25], which was recently developed to solve the original optimization problem, Equation (1). Noticing that $\mathbf{W}$ is a constant matrix, we can extend the original APG method to minimize $E_2(\mathbf{L}, \mathbf{S})$. More specifically, substituting $\mathbf{S} \leftarrow \mathbf{W} \circ \mathbf{S}$ and $\mathbf{P} \leftarrow \mathbf{W} \circ \mathbf{P}$, E_2 becomes

$$E_3(\mathbf{L}, \mathbf{S}) = \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 + \frac{1}{2\mu} \|\mathbf{P} - \mathbf{W} \circ \mathbf{L} - \mathbf{S}\|_F^2 . \tag{3}$$

The only difference between E_3 and E_1 is the entry-wise weighting of $\mathbf{L}$ in the third term. Since such a difference does not change the conditions under which APG is applied, i.e., the cost function consists of a non-smooth convex function $g(\mathbf{L}, \mathbf{S}) = \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1$ and a smooth convex function $f(\mathbf{L}, \mathbf{S}) = \frac{1}{2\mu} \|\mathbf{P} - \mathbf{W} \circ \mathbf{L} - \mathbf{S}\|_F^2$ with its gradient $\nabla f(\mathbf{L}, \mathbf{S})$ Lipschitz continuous, the APG algorithm minimizing E_1 can be directly extended to minimize E_3. For details see Algorithm 1, where $\mathcal{S}_\tau(\mathbf{Z})$ is the entry-wise shrinkage operator on a matrix $\mathbf{Z}$, i.e., $\mathcal{S}_\tau(z) = \mathtt{sgn}(z) \max(|z| - \tau, 0)$ for any element z of $\mathbf{Z}$ and $\mathtt{sgn}(z)$ is the sign of z. As in the original APG [21][25], a continuation technique is applied in Algorithm 1 to reduce the number of necessary iterations by varying μ, i.e., starting from a large initial value μ_0 and then geometrically decreasing by ρ over iterations until reaching the floor $\overline{\mu}$.

The main difference between Algorithm 1 and the original APG one is in Steps 2, 6, and 7, where $\mathbf{W}$ plays its role in the recovery of the low-rank matrix $\mathbf{L}$. Note that the output $\mathbf{S}$ is the weighted sparse matrix. The final $\mathbf{S}$ can be easily estimated from the difference between the input $\mathbf{P}$ and the output $\mathbf{L}$ [29].

Algorithm 1 APG method to minimize Equation (2)

Input: $\mathbf{P}, \mathbf{W}, \lambda$.

$\mathbf{L}_0 = \mathbf{L}_{-1} = \mathbf{0}; \mathbf{S}_0 = \mathbf{S}_{-1} = \mathbf{0}; t_0 = t_{-1} = 1; 0 < \overline{\mu} < \mu_0; 0 < \rho < 1;$

$\mathbf{P} \leftarrow \mathbf{W} \circ \mathbf{P};$

while not converged **do**

$\mathbf{Y}_k^L = \mathbf{L}_k + \frac{t_{k-1}-1}{t_k}(\mathbf{L}_k - \mathbf{L}_{k-1});$

$\mathbf{Y}_k^S = \mathbf{S}_k + \frac{t_{k-1}-1}{t_k}(\mathbf{S}_k - \mathbf{S}_{k-1});$

$\mathbf{G}_k^L = \mathbf{Y}_k^L - \frac{1}{2}\mathbf{W} \circ (\mathbf{W} \circ \mathbf{Y}_k^L + \mathbf{Y}_k^S - \mathbf{P});$

$\mathbf{G}_k^S = \mathbf{Y}_k^S - \frac{1}{2}(\mathbf{W} \circ \mathbf{Y}_k^L + \mathbf{Y}_k^S - \mathbf{P});$

$(\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}) = svd(\mathbf{G}_k^L), \mathbf{L}_{k+1} = \mathbf{U}\mathcal{S}_{\mu_k/2}(\mathbf{\Sigma})\mathbf{V}^{\mathrm{T}};$

$\mathbf{S}_{k+1} = \mathcal{S}_{\lambda\mu_k/2}(\mathbf{G}_k^S);$

$t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2}; \mu_{k+1} = \max(\rho\mu_k, \overline{\mu});$

$k \leftarrow k + 1;$

end while

Output: $\mathbf{L} \leftarrow \mathbf{L}_k, \mathbf{S} \leftarrow \mathbf{S}_k$.

IV. Computing the weighting matrix

The soft-weighting matrix $\mathbf{W}$ is a required input to the proposed method (see Algorithm 1). The basic procedure to generate $\mathbf{W}$ is as follows. First, the candidate impulse noise locations in an image patch are estimated by any of the methods suggested below to obtain a binary weighting matrix $\overline{\mathbf{W}}_0$, in which each entry is set to 1 at the initially estimated impulse noise pixels and 0 elsewhere. Then $\overline{\mathbf{W}}_0$ is convolved with a un-normalized 2D Gaussian operator $G(i,j) = \exp\{-\frac{1}{2\sigma_0^2}(i^2+j^2)\}$, to generate a soft version $\overline{\mathbf{W}}$, where entries with higher value than 1 are set to 1. The final weighting matrix, $\mathbf{W}$, is set to $\mathbf{1} - \overline{\mathbf{W}}$, where $\mathbf{1}$ is a matrix with every entry equal to 1. Consequently, the entries of $\mathbf{W}$ at or near the initially estimated impulse noise locations will have smaller values than elsewhere.

In practice, the binary matrix $\overline{\mathbf{W}}_0$ can be generated by any existing impulse noise detectors (e.g., ROAD [13] or ROLD [11]), or even by our low-rank matrix recovery framework by setting $\mathbf{W} = \mathbf{1}$. This is because our framework (when $\mathbf{W} = \mathbf{1}$) also recovers the sparse matrix $\mathbf{S}$ representing the initial spatial distribution of impulse noise. Note that the denosing performance of the proposed method is robust to occasional errors in the estimated $\mathbf{W}$: If certain noise-free pixels occasionally have smaller weights in $\mathbf{W}$, such lower-weighted noise-free pixel values won't be changed too much in the final denoised low-rank patch because the rank of the patch

(a) (b) (c)

Fig. 2: Effect of characteristic orientation on low-rank patch approximation. (a) A synthetic image. (b) A low-rank (rank 15) approximation of a 41×41 image patch around the image center. (c) The low-rank (rank 1) approximation of an oriented image patch around the same point.

is already small without changing these pixel values, while changing pixel values will probably increase the value of the second cost term in Equation (2).

V. Characteristic orientation

One potential issue in denoising methods is edge blurring and loss of sharpness. Patch orientation affects the result of our rank-based method. For example, even a patch with a simple pattern may have a high rank (Figure 2, patch in blue rectangle). In this case, the low-rank approximation of the patch will blur the sharp edge (Figure 2b). Instead, if we can find a low-rank patch (Figure 2, patch in yellow rectangle) by rotating around the target image point, the low-rank approximation of this patch will more likely preserve the sharpness of the edges (Figure 2c). Similar observations apply for patches with other texture patterns like corners, and our experiments (Figure 8) show the importance of characteristic orientation in denoising.

Based on the assumption that the optimally oriented patch is low-rank, we expect that the difference between the oriented patch and its low-rank approximation will be minimum at the optimal ('characteristic') orientation. To compute the latter, let $\mathbf{P}(\theta)$ denote an oriented $m \times n$ image patch at a given image position, rotated anticlockwise by an orientation angle θ with respect to the image row direction, and $\tilde{\mathbf{P}}(\theta)$ the low-rank approximation (to a fixed quality level) of $\mathbf{P}(\theta)$. Then the characteristic orientation at the current image position can be estimated as

$$\hat{\theta} = \arg \min_{\theta \in [0,\pi]} d(\theta) = \arg \min_{\theta \in [0,\pi]} \frac{1}{mn} \left\| \mathbf{P}(\theta) - \tilde{\mathbf{P}}(\theta) \right\|^2 , \tag{4}$$

where $\|\cdot\|$ is a matrix norm (we use Frobenius), and θ is restricted in the range $[0, \pi]$ because any pair of θ and $\theta + \pi$ will lead to the same norm value.

In the above, the quality level of the low-rank approximation should be defined and a threshold introduced. However we found that a basic, efficiently computed rank-1 approximation of $\mathbf{P}(\theta)$ to represent $\tilde{\mathbf{P}}(\theta)$, i.e., every column in $\tilde{\mathbf{P}}(\theta)$ is the mean of all columns in $\mathbf{P}(\theta)$, leads to very good overall results especially in preserving edges. Such a simple, threshold-free approximation proved effective enough to find reliably the characteristic orientation for each image patch (Section VI-C).

VI. Experiments

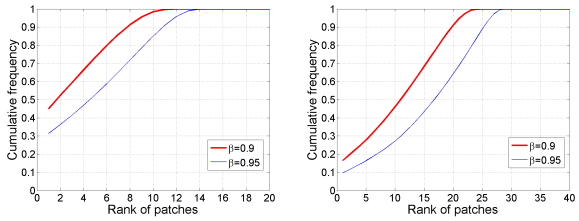
A. Experimental set-up

Our APG algorithm (Algorithm 1) was implemented by modifying the public MATLAB source code for the original APG [21]. Similar to the parameter settings in [5], $\mu_0 = 0.99\|\mathbf{W} \circ \mathbf{P}\|_2$ and $\overline{\mu} = 10^{-9}\mu_0$. For an $m \times n$ image patch $\mathbf{P}$, $\lambda = 1/\sqrt{\max(m, n)}$. The maximum iteration number is set to 200. To find the characteristic orientation $\hat{\theta}$, a simple uniform sampling method was adopted with sampling interval $\frac{\pi}{36}$. σ_0 and the window size for the un-normalized Gaussian filter G were set to $\frac{1}{36}\min(m, n)$ and $\frac{1}{6}\min(m, n)$ respectively. When using the initially estimated sparse matrix $\mathbf{S}$ to generate $\overline{\mathbf{W}}_0$, the candidate noise locations are the entries where the absolute value is larger than a threshold, determined adaptively based on the expected sparsity level of the noise (i.e., number of noisy pixels over total pixel number). We use a sparsity level of 0.05.

All the tests were performed using Matlab R2010a running on an Intel Core i7-2600K 3.40GHz PC with 8.0GB RAM. For an image with size 640×480 pixels, and patches with size 41×41 pixels and 50% overlapped by neighboring patches along both directions, it takes approximately 3 minutes in all to generate the denoised image. Since each image pixel is often covered by multiple patches, the final denoised value at each pixel in the image is averaged from the corresponding denoised values in these multiple patches.

B. Low-rank prior in single patches

We illustrate the experimental evidence of the low-rank prior for small noise-free image patches. In practice, due to noise, an image patch as a matrix is seldom low-rank. However, if the assumption of low-rank prior is true, the column-wise signal variation in an image patch



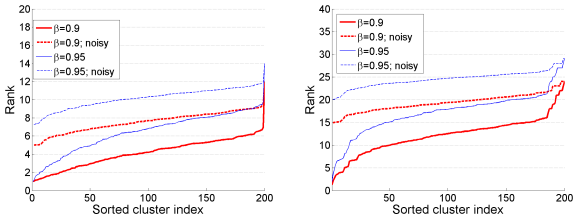
(a) (b)

Fig. 3: Low-rank prior in image patches with size (a) 21×21 (from Caltech256) and (b) 41×41 pixels (from SceneCategory15).

should be mostly preserved in a much lower-dimensional space, and the low-rank approximation should preserve meaningful textural details. Such predictions are confirmed empirically from our tests with the public datasets Caltech256 [14] and SceneCategory15 [18], and around $450,000$ oriented (i.e., rotated to their characteristic orientations) patches with sizes 21×21 and 41×41 pixels were generated from each dataset by uniform sampling in each image.

The first test explores statistically the order of the low-rank approximation needed to preserve, at a given level β, the column-wise signal variation in each $m \times n$ image patch. Every patch is first decomposed by SVD and the minimum number $\hat{l}$ of singular values necessary to preserve the predefined level of signal variation is determined by formula $\hat{l} = \arg\min_{J \in [1, \min(m,n)]} \{\beta < \sum_{j=1}^{J} \sigma_j / \sum_{j=1}^{\min(m,n)} \sigma_j\}$, with σ_j being the j-th largest singular value. Different patches may have different rank values $\hat{l}$. A rank histogram can be easily generated recording the frequency of patches with a particular rank value. The cumulative rank histogram in Figure (3a) (thinner blue line) shows that when preserving 95% of the column-wise signal variations, about 90% image patches have low-rank approximations with rank equal or smaller than 11. When $\beta = 0.9$, more than 98% image patches have low-rank approximations with rank equal or smaller than 10. Similar results were obtained for various sizes in both datasets (Figure 3b). This shows that most oriented patches can be approximated by their low-rank versions which keep most signal variations.

The second test shows that low-rank patch approximations can preserve textural details. A fast k-means method [2] was applied to cluster the patches of a fixed size into 200 clusters using the datasets above. The average $\hat{l}$ over all patches was computed within each cluster. Figure 4 (solid



(a) (b)

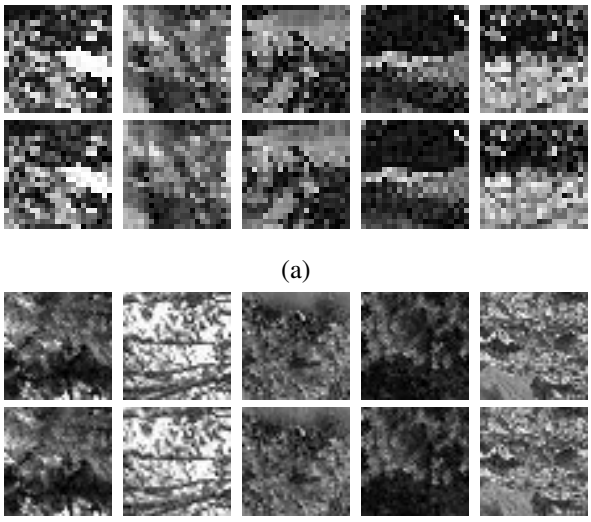
Fig. 4: Average rank value of each cluster for image patches with size (a) 21×21 (from Caltech256) and (b) 41×41 pixels (from SceneCategory15).

curves) shows the sorted average rank values for all the clusters. Consistent with the first test, when $\beta = 0.95$, most clusters have average rank values less than 10 for 21×21 patches and less than 20 for 41×41 patches. The small subset of clusters with larger average rank values often corresponds to the patches with more complex visual appearance or patterns. Within each such cluster, the highest-rank (largest $\hat{l}$) image patch was chosen to represent the most complex visual pattern. Figure 5a lists such image patches and the corresponding low-rank approximations. It can be observed that, even for the patches with most complex texture patterns, the textural details have all been preserved in the low-rank approximations. Similar observations have been found for the 41×41 image patches (Figure 5b). Such observations are not trivially true for (at least some) un-oriented patches (see Figure 2b).

In addition, when adding RVIN to the image patches by corrupting 5% pixels in each patch, Figure 4 (dashed curves) also shows that the average rank values increased. This suggests not only that the noise-free image patches are low-rank, but also that the noisy patches have higher ranks at a predefined level β. The proposed low-rank matrix recovery framework just makes use of this observation for removing noise from image patches.

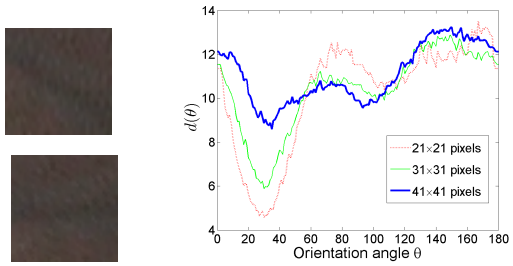
C. Characteristic orientation

The first test here checks whether the determination of characteristic orientation is invariant to changes in patch sizes. As an example, we use an image patch (Figure 6a top) from a noisy and low-contrast underwater hydrocolonoscopy image (Figure 9a; see more information about hydrocolonoscopy images in Section VI-F). Figure 6 shows that for a large range of patch sizes



(a) (b)

Fig. 5: Image patches for the 5 clusters with highest average rank values, with patch size (a) 21×21 pixels (top: original; bottom: rank-10 approximation) and (b) 41×41 pixels (top: original; bottom: rank-20 approximation).



(a) (b)

Fig. 6: Determination of characteristic orientation. (a) An original low-contrast 41×41 image patch (top) and the oriented version (bottom) (better view on monitor). (b) $d(\theta)$ over all possible orientation angles with three patch sizes.

(e.g., 21×21 to 41×41), the estimated characteristic orientations are almost the same (the minima of all curves occur at the same angle). This invariance to patch sizes is especially beneficial to denoise images with textures at different scales.

The second test checks whether the orientation determination is robust to noise in patches.

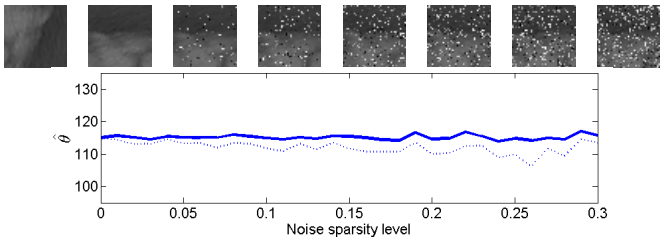


Fig. 7: Robustness of characteristic orientation determination. Top row (left to right): original image patch, oriented patches with noise sparsity level at 0.00, 0.05, 0.10, 0.15, 0.20, 0.25, and 0.30. Bottom: estimated characteristic orientation with varying noise sparsity levels, with mean (solid curve) and standard deviation (dotted curve) values from 10 runs.

Given an image patch (Figure 7 top left), Figure 7 shows that the estimate of the characteristic orientation is robust to RVIN noise, even if 30% of image pixels have been damaged by RVIN. Figure 8 further confirms the robustness of the orientation determination to noise even for high-textured and low-textured image patches. Of course, the robustness would degrade with increasing level of RVIN particularly for high-textured patches, because pseudo-textures caused by the increasing RVIN will increase the uncertainty during characteristic orientation determination, as shown in Figure 8b.

The third test compares the proposed method with the well-known 'SIFT-orientation' method [22], where the dominant gradient direction was determined based on the weighted histogram of gradient orientations. The image was smoothed twice by a Gaussian with standard deviation 1.6 and window size $6 * 1.6$ pixels; the gradient with 4 pixels spacing along both dimensions was then computed at every location. Different Gaussian and spacing parameters were tried with results similar to Figure 9b. Compared to the results by the proposed method (Figure 9c), the characteristic orientations by the SIFT-orientation method are often not precisely orthogonal to the dominant edge in each image patch, probably due to noise. In the denoised result, based on the proposed matrix recovery framework, some edges have been blurred when using SIFT-orientation method (Figure 9e). In comparison, the sharpness of edges has been preserved by the proposed method (Figure 9f). While sampling error is introduced in the oriented patches by rotating the original patches, the consequent loss of image quality appears much smaller compared to the enhancement due to noise removal from the oriented patches. This is supported

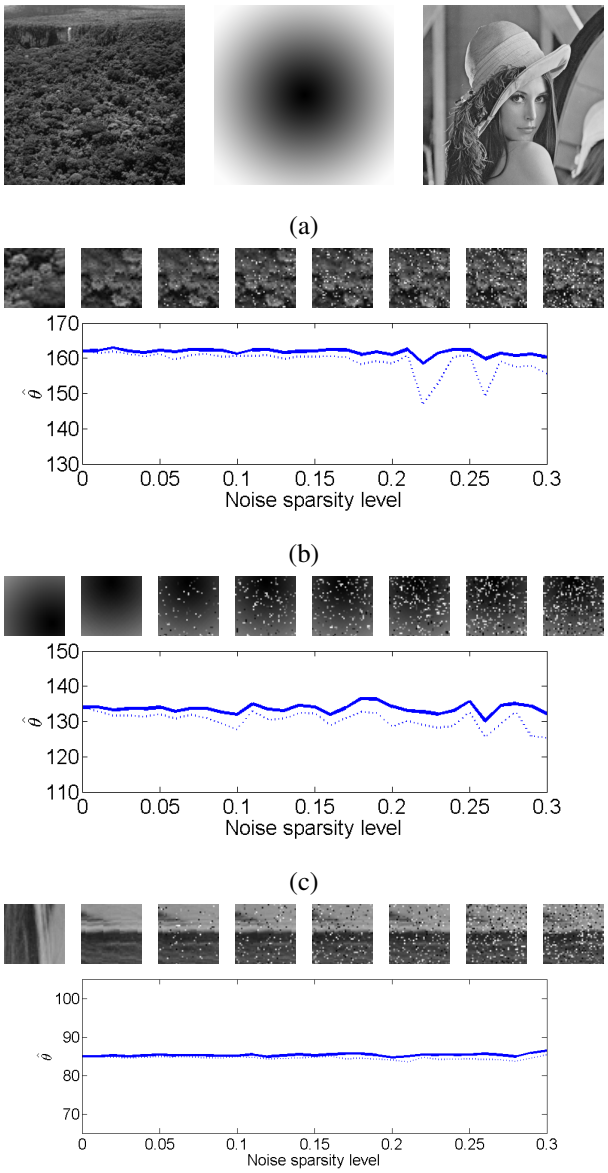


Fig. 8: Robustness of characteristic orientation determination at high-textured and low-textured image regions. (a) A real high-textured image (left), a synthetic low-textured image (middle), and the Lena image (right). (b-d) Orientation determination for a randomly selected image patch from the high-textured image, the low-textured image, and the Lena image.

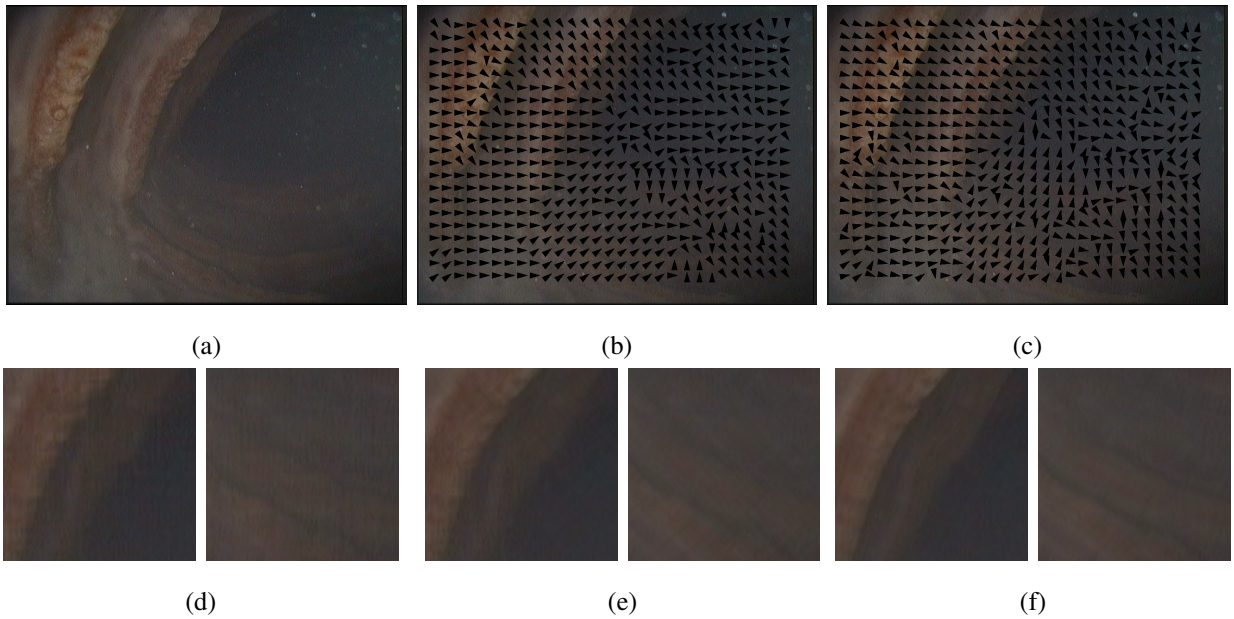


Fig. 9: Characteristic orientation and its effect on denoising. (a) The original image. The orientation (by arrow) determined by (b) the SIFT-orientation method and (c) the proposed method for 41×41 patches. Two cropped denoised image regions (d) without orientation determination, (e) using the SIFT-orientation method, and (f) using the proposed method.

by Figures 9d-9f, which show that both orientation methods perform better than the one not using orientation (Figure 9d).

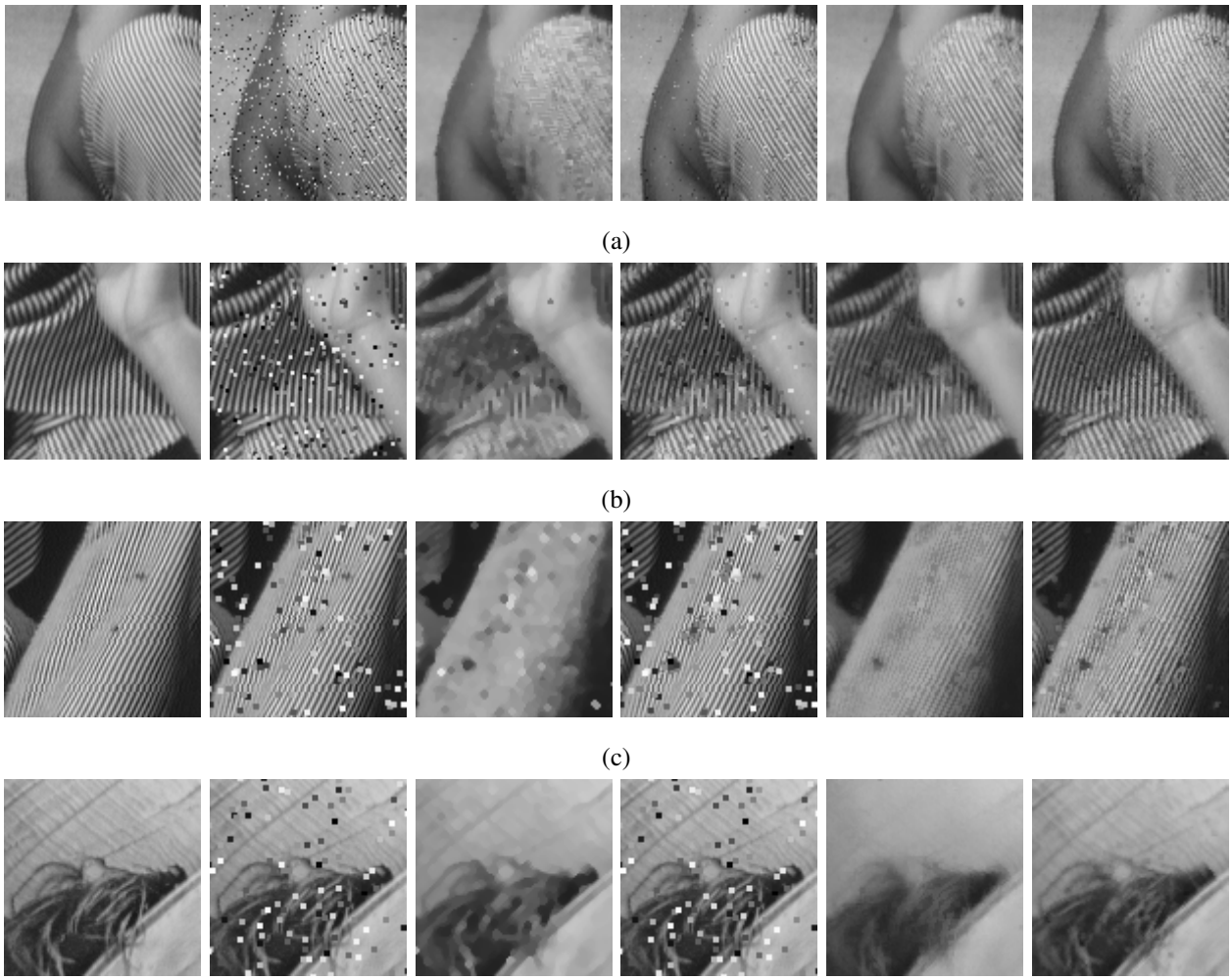
D. Detection and removal of impulse noise

To evaluate our method quantitatively, RVIN with different sizes (e.g., 1×1 to 4×4 pixels) at a particular sparsity level (i.e., 0.1) was added to images respectively to generate the noisy images. For example, for RVIN with size 2×2 pixels, 2.5% (i.e., $0.1/(2 \times 2)$) of pixel locations were randomly selected from a noise-free image, and then at each selected location, a 2×2 RVIN was generated, overall making 10% of image regions become noisy. Although results with only two well-known images ('Barbara' and 'Lena') are used to demonstrate the performance due to limited space, similar results have been obtained for other widely-used standard images (e.g., Baboon, Finger, Bridge, Hill, Man, Boat). The denoising performance was measured by standard peak signal-to-noise ratio (PSNR).

Five denoising methods were chosen for comparison: the median filtering as the baseline method, the NS-LI method [7], the ROLD-EPR method [11], the multi-patch low-rank matrix recovery method (MPLR) [16], and the proposed method without applying the weighting matrix $\mathbf{W}$ (henceforth 'Proposed $\backslash\mathbf{W}$'). For median filtering, 20 iterations were run with window size 3×3 pixels. The PSNR was computed after each iteration and the highest PSNR value reported. Similarly for ROLD-EPR, the maximum iteration was set to 20, the window size to 5×5 and all the other parameters were set as suggested in [11]. The highest PSNR over all iterations is reported. For NS-LI, all the parameters were set as suggested in [7]. Similarly for MPLR, all the parameters were set as suggested in [16], except that four different patch sizes (i.e., $n\times n$, $n\in\{4,8,16,32\}$) were tried and 10 additional similar patches were searched (to generate a $n^2\times 11$ matrix) across the whole image when denoising each image patch. The highest PSNR over the different patch sizes is reported. For our method, the patch size was fixed to 31×31, the highest PSNR is reported over different $\lambda=s/\sqrt{31}$, where $s\in\{1.0,1.3,1.6,2.0\}$. For each method, 10 runs were performed, with PSNR standard deviation around 0.1.

Table I (last two rows) shows that the proposed method consistently performs comparably well or better than 'Proposed $\backslash\mathbf{W}$' for smaller-size RVIN (26.45 and 26.40 for 'Barbara' 3×3 RVIN are not significantly different). The substantial improvement in PSNR for large-size (i.e., 4×4) RVIN is due to the effect of the weighting matrix $\mathbf{W}$. Table I also shows that the proposed method performs better than the other methods with non-pointwise RVIN (i.e., 3×3 and 4×4). This is probably due to the limited ability to detect and remove large-size RVIN by the NS-LI method and the ROLD-EPR method, and the oversmoothing by median filtering and MPLR, as demonstrated in Figures 10c and 10d. Note that the parameters of the NS-LI method have not been exactly optimized in the tests, therefore the performance of NS-LI would be better than its performance reported here. For the image with more textured regions (i.e., 'Barbara'), the proposed method also performs better even when RVIN is smaller (i.e., 1×1 and 2×2). Figure 10a and 10b show that noticeable noise still remains after denoising by ROLD-EPR, and the regular patterns have been blurred by Median and MPLR. Consistent with PSNR assessment, the proposed method gives the best visual quality.

For the image with less textured regions (i.e., 'Lena'), ROLD-EPR performs best for RVIN of size 2×2 based on the PSNR evaluation. However, detailed quantitative analysis found that about 20% of the RVIN was still present in the denoised image by ROLD-EPR (Figure 11b). In



(a) (b) (c) (d)

Fig. 10: Part of denoised images ((a)-(c) from Barbara, and (d) from Lena) from different methods with noise size (a) 1×1, (b) 2×2, and (c) and (d) 3×3 pixels. From left to right in column: original clean patch, noisy patch, denoised result by Median, ROLD-EPR, MPLR, and the proposed method.

comparison, only 8% of the RVIN was present in the denoised image by the proposed method. This indicates that the higher PSNR by ROLD-EPR is due largely to keeping the non-corrupted pixels from changing rather than removing more RVIN. We analyzed carefully the noisy spots remaining in the denoised images after applying our method (e.g., near the nose and the mouth in Figure 11d). We found that such noise coincidentally appear densely either in a row or a

TABLE I: PSNR from different methods for different sizes of RVIN.

	Barbara				Lena			
	1×1	2×2	3×3	4×4	1×1	2×2	3×3	4×4
Noisy image	19.02	19.09	19.00	18.93	19.43	19.52	19.56	19.30
Median	24.90	23.90	23.28	18.94	33.64	30.78	28.65	20.37
NS-LI	28.89	19.09	18.96	18.92	32.24	19.59	19.56	19.32
ROLD-EPR	29.48	26.80	18.99	18.90	35.89	**33.58**	19.63	19.34
MPLR	28.60	25.95	24.06	23.61	**37.18**	31.34	27.28	26.58
Proposed $\setminus \mathbf{W}$	29.69	28.13	**26.45**	23.99	33.32	30.99	28.65	25.15
Proposed	**30.69**	**28.43**	26.40	**25.34**	34.44	31.12	**28.90**	**27.30**

column in the oriented patch, and at the same time such row or column of noise is aligned with the edge structure in direction within the region (e.g., see the noise along the edge of nose in Figure 11d). In this case, removing the row or column of the noise from the patch would probably not reduce the cost of the first cost term (i.e., the rank of the oriented patch) but increase the costs of the last two cost terms in Formula 2. As a result, the proposed method only modified the row (or column) of noise to make them have similar intensity values rather than completely removed them. This may also explain why the proposed method could have different performance in different image regions. Compared to nose and mouse regions, the shoulder region has no edge-like structures and more homogeneous. If some noise appears densely along a line in this region, removing such noise from the patch will more likely reduce the cost of the first cost term, because a homogenous patch often has significantly lower rank than a (even edge-like) textured patch. In addition, although MPLR and the proposed method have similar PSNR values (31.34 vs. 31.12) for the 2×2 RVIN, MPLR often over-smooth image regions (Figure 11c). In general, it seems that PSNR alone may not be sufficient to represent the visual quality of denoised images in the presence of RVIN noise. It remains as future work to explore novel quantitiave measurement for RVIN noise. Finally, the superior performance by MPLR when RVIN size is 1×1 simply confirms previous results when combining the non-local idea with low-rank matrix recovery [16].

While the above analysis is based on a specific noise sparsity level (i.e., 0.1), similar results have been obtained in a large range of sparsity levels. As an example, Table II shows that at the sparsity levels 0.05 and 0.15, the proposed method outperforms all other methods based on the

TABLE II: PSNR at different sparsity levels.

	Sparsity=0.05				Sparsity=0.15			
	1×1	2×2	3×3	4×4	1×1	2×2	3×3	4×4
Noisy image	21.97	22.01	21.47	21.69	17.40	17.34	17.35	17.49
Median	25.17	24.42	23.74	20.94	24.57	23.46	22.59	17.75
NS-LI	31.90	21.93	21.43	21.62	26.47	17.38	17.32	17.46
ROLD-EPR	32.08	29.16	21.44	21.64	27.94	25.43	17.38	17.48
MPLR	29.02	27.41	26.16	24.19	27.41	25.36	24.30	23.02
Proposed	**32.54**	**30.98**	**28.56**	**27.39**	**29.06**	**27.53**	**25.86**	**24.39**

TABLE III: PSNR for mixed sizes (from 1×1 to 3×3) of RVIN.

	Noisy image	Median	NS-LI	ROLD-EPR	MPLR	Proposed
Barbara	18.98	23.72	20.39	24.12	25.84	28.42
Lena	19.49	30.22	21.12	28.08	31.48	30.58

image 'Barbara', consistent with the results at the sparsity level 0.1 (Table I).

In addition, in real applications, noise with different sizes may appear simultaneously in images. To test the performance of the proposed method in this scenario, RVIN with three different sizes (i.e., 1×1, 2×2, 3×3) were mixed together, with each size of RVIN generated at a sparsity level 0.033 (so overall the noise sparsity level is 0.1). In this case, although the sizes of most RVIN are small (i.e., 1×1 or 2×2), consistent with the single-size noise detection and removal, the proposed method outperforms others on 'Barbara' and is a very close second on 'Lena' (Table III).

E. Robustness for weighting matrix

As discussed in Section IV, the overall denoising performance is expected to be robust to possible errors during the generation of the weighting matrix $\mathbf{W}$. To confirm this analysis experimentally, an initial estimate (i.e., the binary matrix $\overline{\mathbf{W}}_0$) with errors was generated as follows. First, 10% pixels of each noise-free image were replaced by 2×2 RVIN blobs. The initial estimate $\overline{\mathbf{W}}_0$ without errors can be generated based on the ground-truth RVIN positions (corresponding to '0' in X-axis in Figure 12). To generate an noisy $\overline{\mathbf{W}}_0$, erroneously estimated 2×2 RVINs were generated by shifting each ground-truth 2×2 RVIN randomly to one of four

(a) Noisy image (19.52) (b) ROLD-EPR (33.58) (c) MPLR (31.34) (d) Proposed (31.12)

Fig. 11: Visual quality comparison between different methods (PSNR value in bracket) for RVIN of size 2×2. Note the visible remaining RVIN especially in smooth regions (e.g., shoulder, background) in (b) and over-smoothed regions (e.g., hat) in (c).

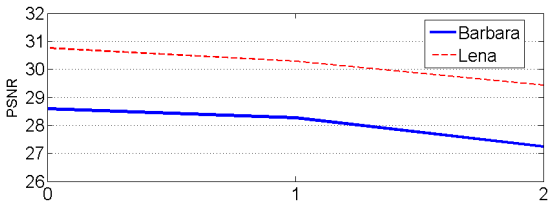


Fig. 12: Robustness of weighting matrix. See text for details.

directions (left, right, up, down) by one ('1' along X-axis in Figure 12) or two ('2' along X-axis) pixels. Considering the size (2×2) of each RVIN, shifting each ground-truth RVIN by one pixel indicates that half of initial noise positions in the initial estimate $\overline{\mathbf{W}}_0$ are incorrect, and shifting by two pixels indicates that all initial noise positions are incorrect. Figure 12 shows that there is no significant change in denoising performance when half of the initial RVIN positions are incorrect, and a limited decrease even when all initial positions are incorrect, together demonstrating the robustness of the proposed method to errors in the initial weighting matrix.

F. Particle removal in hydrocolonoscopy images

Hydrocolonoscopy is considered as a promising replacement of traditional air colonoscopy, as limiting patient discomfort [19]. We tested our noise removal technique on hydrocolonoscopy images in the framework of the CODIR research project (Colonic Disease Investigation by Robot Hydro-Colonoscopy) [1]. A hydrocolonoscopy procedure was simulated with a latex phantom of a human colon section immersed in water in a tank (Figure 13). The colon phantom (size, shape, colour and texture on the internal wall) was manufactured by reverse-engineering MRI data of a human colon. The tank could be darkened completely by a cover to prevent visual artifacts created by natural light. About 1000 hydrocolonoscopy images of the colon phantom wall were extracted from a video captured by a state-of-the-art colonoscope (carrying its own illumination) mounted on a movable rod to simulate the interventional motion. Water turbidity and floating particles, simulating noisy interventional conditions, were created by diluting controlled amounts of dispersion paint in the water. Particles with various sizes (maximumly 15×15 pixels) were suspended in water to create non-pointwise RVIN.

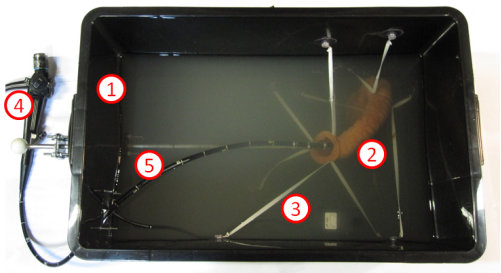


Fig. 13: Experimental set up for hydrocolonoscopy image acquisition with turbid water. The set up consists of a (1) tank, (2) the yelowish colon phantom, (3) the white suspension to maintain a constant position of the colon, (4) an colonoscope and (5) a movable rod to position the colonoscope.

The proposed denoising method removed effectively particle noise in hydrocolonoscopy images. Figure 14 displays a representative example: an hydrocolonoscopy image and the denoised result with the MPLR and the proposed method. The patch size is 41×41 for the proposed method and 16×16 for the MPLR method in order to remove large-size particles. MPLR oversmoothed the image with larger particles partially remaining (Figure 14c and 14d). The 'Proposed $\backslash \mathbf{W}$'

can preserve sharpness but cannot remove large-size particles effectively (Figure 14e and 14f). The proposed method can remove the particles effectively while preserving the sharpness of the edges (Figure 14g and 14h). These results are representative of those obtained with all our hydrocolonoscopy images.

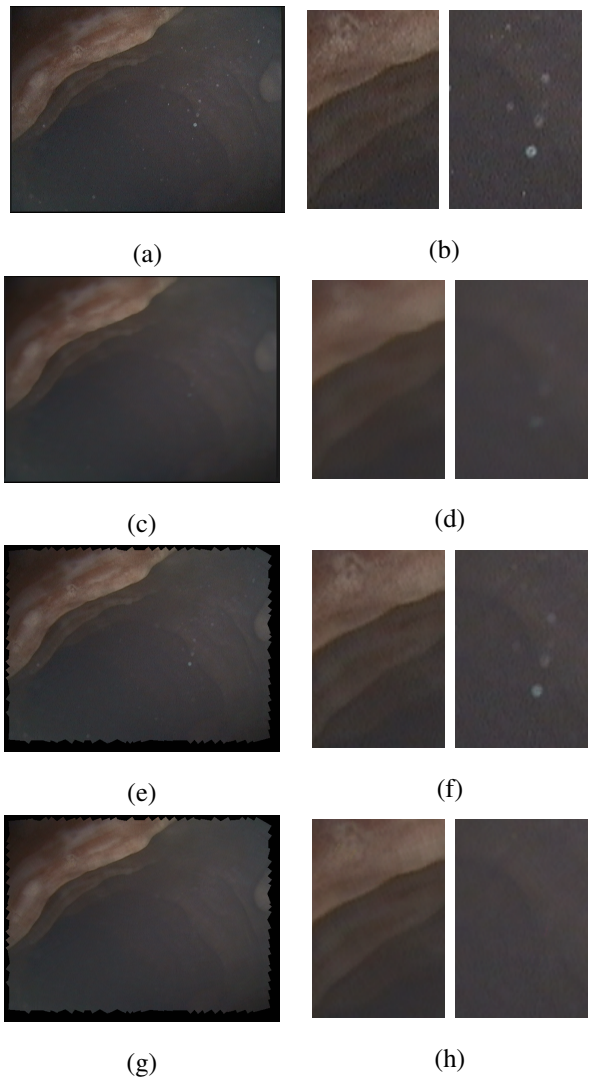


Fig. 14: Removal of particles suspended in a hydrocolonoscopy image. (a) The original noisy image. (c) Denoised image by MPLR with patch size 16×16 pixels. (e) Denoised image by the proposed method without $\mathbf{W}$ (dark boundary regions not denoised). (g) Denoised by the proposed method. Right column (b,d,f,h): Two image patches cropped from each of the images in the left column.

G. Method limitations

Like every other denoising method, the proposed method is limited by its assumptions. First, although most image patches are low-rank after being appropriately oriented (Section VI-B), there are still a few patches which do not satisfy the low-rank assumption. For these patches, denoising with the proposed method will inevitably over-smooth the original patches. Second, for images or image regions with fine textures, there is a trade-off between removing RVIN and preserving fine texture details. This trade-off can be realized by tuning the parameter λ, with smaller λ emphasizing RVIN removal and larger λ emphasizing fine texture preservation. Figure 15 shows an example of denoising the image 'Bridge'. Smaller λ results in a smoother but less detailed denoised image (Figure 15b) than larger λ (Figure 15c). Third, the proposed method is essentially low-level, and does not consider any high-level (e.g., object level) information. If an image contains small-scale objects which look like non-pointwise RVIN, they could be removed as RVIN by the proposed method. For example, the two bright dots in the 'Bridge' image were preserved in the denoised image (circled in Figure 15c) when λ is relatively large, but incorrectly removed when λ is relatively small (Figure 15b). In addition, the proposed method requires the rotation of each image patch by an appropriate orientation angle before denoising. Such rotation may introduce sampling errors. However, if rotating patches help remove noise more effectively than without rotation, the loss due to sampling error will be smaller than the gain of the better noise removal. This is supported by the experimental results (e.g., Figure 9d vs. 9f).

VII. Conclusions

This paper has introduced a low-rank prior for small oriented (rotated by a characteristic orientation angle) noise-free image patches. The low-rank prior suggests that a single patch can be effectively denoised within a low-rank matrix recovery framework. Without resorting to other similar patches, the single-patch method can effectively remove non-pointwise RVIN within a generalized low-rank matrix recovery framework, and encode the initial estimation of noise locations effectively. Experimental results show the better performance of the proposed approach over several methods, especially for non-pointwise RVIN. Removing random Gaussian noise and video denoising will be explored as future work.

(a) Noisy image (b) Denoised image (small λ) (c) Denoised image (larger λ)

Fig. 15: Denoised 'Bridge' by the proposed method with two different λ. More fine textures (e.g., top-right region) are preserved with larger λ, and RVINs are more completely removed with small λ. Small object parts (circled in c) could be incorrectly removed with small λ.

ACKNOWLEDGMENT

This work is funded by 2011-2016 EU FP7 ERC project "CODIR: colonic disease investigation by robotic hydrocolonoscopy", collaborative between the Universities of Dundee (PI Prof Sir A Cuschieri) and Leeds (PI Prof A Neville). Thanks to Dr A Alazmani and Mr S Manivannan for hydrocolonoscopy video capture, and Prof P Lin and CVIP members for comments and discussions.

REFERENCES

[1] "CODIR: colonic disease investigation by robot hydro-colonoscopy," 2014, http://www.codir.org.

[2] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *SODA*, 2007.

[3] E. Besdok, P. Civicioglu, and M. Alci, "Impulsive noise suppression from highly corrupted images by using resilient neural networks," *ICAISC*, vol. 3070, pp. 670–675, 2004.

[4] A. Buades, B. Coll, and J. Morel, "Image denoising methods. a new nonlocal principle," *SIAM Review*, vol. 52, no. 1, pp. 113–147, 2010.

[5] E. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *JACM*, vol. 58, no. 3, 2011.

[6] T. Chen and H. Wu, "Adaptive impulse dectection using center-weighted median filters," *SPL*, vol. 8, no. 1, pp. 1–3, 2001.

[7] P. Civicioglu, "Removal of random-valued impulsive noise from corrupted images," *TCE*, vol. 55, no. 4, pp. 2097–2104, 2009.

[8] P. Civicioglu and M. Alci, "Edge detection of highly distorted images suffering from impulsive noise," *AEU-IJEC*, vol. 58, no. 6, pp. 413–419, 2004.

[9] K. Dabov, V. Katkovnik, R. Foi, K. Egiazarian, and S. Member, "Image denoising by sparse 3D transform-domain collaborative filtering," *TIP*, vol. 16, no. 8, pp. 2080–2095, 2007.

[10] W. Dong, X. Li, L. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structure clustering," in *CVPR*, 2011.

[11] Y. Dong, R. Chan, and S. Xu, "A detection statistic for random-valued impulse noise," *TIP*, vol. 16, no. 4, pp. 1112–1120, 2007.

[12] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *TIP*, vol. 15, no. 12, pp. 3736–3745, 2006.

[13] R. Garnett, T. Huegerich, C. Chui, and W. He, "A universal noise removal algorithm with an impulse detector," *TIP*, vol. 14, no. 11, pp. 1747–1754, 2006.

[14] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007, cNS-TR-2007-001.

[15] C. Guillemot and O. L. Meur, "Image inpainting : Overview and recent advances," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 127–144, 2014.

[16] H. Ji, S. Huang, Z. Shen, and Y. Xu, "Robust video restoration by joint sparse and low rank matrix approximation," *SIAM J. Imaging Sci.*, vol. 4, no. 4, pp. 1122–1142, 2011.

[17] V. Katkovnik, A. Foi, K. Egiazarian, and J. Astola, "From local kernel to nonlocal multiple-model image denoising," *IJCV*, vol. 86, no. 1, pp. 1–32, 2010.

[18] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006.

[19] F. Leung, A. Amato, C. Ell, and S. F. et al., "Water-aided colonoscopy: A systematic review," *Gastrointestinal Endoscopy*, vol. 76, no. 3, pp. 657–666, 2012.

[20] X. Liang, X. Ren, Z. Zhang, and Y. Ma, "Repairing sparse low-rank texture," in *ECCV*, 2012.

[21] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," technical Report UILU-ENG-09-2214.

[22] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[23] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *ICCV*, 2009.

[24] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *TIP*, vol. 17, no. 1, pp. 53–69, 2008.

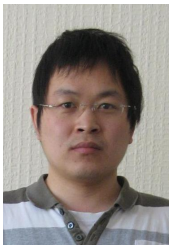
[25] K. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems," *Pacific J. Optimization*, vol. 6, no. 3, pp. 615–640, 2010.

[26] R. Wang and E. Trucco, "Single-patch low-rank prior for non-pointwise impulse noise removal," in *ICCV*, 2013.

[27] Y. Xiao, T. Zeng, J. Yu, and M. Ng, "Restoration of images corrupted by mixed gaussian-impulse noise via $l_1 - l_0$ minimization," *PR*, vol. 44, no. 8, pp. 1708–1720, 2011.

[28] Z. Zhang, A. Ganesh, X. Liang, and Y. Ma, "Tilt: Transform invariant low-rank textures," *IJCV*, vol. 99, no. 1, pp. 1–24, 2012.

[29] Y. Zhou, Z. Ye, and Y. Xiao, "A restoration algorithm for images contaminated by mixed gaussian plus random-valued impulse noise," *J. Vis. Commun. Image R.*, vol. 99, pp. 283–294, 2013.



Ruixuan Wang received BEng and MEng degrees in Automtic Control and Artificial Intelligence from Xi'an Jiaotong University, China, in 1999 and 2002, and a PhD in Computer Vision from the National University of Singapore in 2007. As a postdoctoral researcher in University of Dundee since 2007, he has been involved in various multi-disciplinary research projects, including image browsing, microscopy and colonoscopy image analysis. His research interests include computer vision, medical image analysis, and machine learning.

Markus Pakleppa Markus Pakleppa has received his MSc degree in Microtechnology and Medical Engineering in 2010 from the University of Applied Sciences Gelsenkirchen (Germany). Within his final year project he worked on the development of a measurement device for breath analysis. In 2011 he enrolled in the PhD programme of the School of Engineering, Physics and Mathematics at the University of Dundee under the supervision of Dr Robert Keatch and Dr Jan Bernd Vorstius. He has since become a member of the FP7 ERC funded CoDIR project (colonic disease investigation by robotic hydro colonoscopy) and his PhD work includes the development of a colonoscopy simulator for the evaluation of colonoscopy devices. His research interest includes Medical Engineering, Measurement Technology and Mechatronics.

Emanuele Trucco , FRSA (MSc 1984, PhD 1991), is the Northern Research Partnership Chair of Computational Vision at the University of Dundee, UK, and an Honorary Clinical Researcher at the Ninewells National Health Service (NHS) and University Hospital, Dundee. His research has covered applications of computer vision to manufacturing, subsea robotics, immersive videoconferencing, and medical image analysis. He has managed projects funded by the UK Research Councils, the EU, industry (incl. British Telecom, OPTOS plc, Toshiba) and various charities (incl. British Council, Royal Society, Leverhulme Trust). His main research interests are in retinal image analysis; he co-directs the VAMPIRE initiative (Vessel Assessment and Mesurement Platform for Images of the REtina) growing a software suite enabling efficient semi-automatic analysis of retinal images. VAMPIRE has been used in retinal biomarker studies on, among others, cardiovascular risk, stroke, cognitive performance, neurodegenerative diseases, vascular dementia, and genetics. Further research include vision for robotic hydrocolonoscopy (EU ERC "CODIR") and the analysis of whole-body MR angiographic data (with Toshiba and the UK NHS). Trucco has served regularly on the program or technical committees of all major conferences in computer vision, incl. as program chair for 3 editions of the British Machine Vision Conference. He has served as Editor-in-Chief of the IEE Transactions on Image Processing, and as Associate Editor for various international journals. Trucco has published more than 200 refereed papers and co-authored two textbooks, one of which (with Alessandro Verri) has become an international standard (2,464 citations, Google Scholar, Jan 2015).